Microsoft

.net developer training

Create XML Web Services and Applications with Visual Studio .NET

The .NET Languages

Module 03 Agenda

- Languages
- Visual Basic® .NET
- Visual C#™ .NET
- Visual C++® .NET
- Visual J# .NET

.NET Languages

- The .NET Platform is multi-language
  - Benefits:
    - Make code modules reusable
    - Make API access the same for all languages
    - Let developers use the right language for the task and share the code
  - All .NET languages are first-class players
    - Performance roughly equal
    - Can use all .NET features
- Microsoft is providing:
  - Visual Basic, Visual C++, Visual C#, Visual J# and JScript®

.NET Languages

- Visual Basic .NET
- Visual C# .NET
- Visual C++ .NET
- Visual J# .NET
- JScript .NET
- C
- APL
- Component Pascal
- Haskell
- Standard ML
- COBOL
- RPG
- Perl
- Python
- SmallTalk
- Eiffel
- Mercury
- Oberon
- Mondrian
- Scheme
- Fortran

Choosing a Language

- Languages are true equals
- Language selection
  - Visual Basic or VBScript developer → Visual Basic .NET
  - Visual C++ → Visual C++ (Managed), Visual C#
  - JScript developer → JScript or Visual C#
  - Java developer → Visual J#, Visual C#

Multi Language
Demo

- COBOL, Visual C# and Visual Basic .NET demo

Visual Basic .NET

- .NET integration
  - VBRUN replaced by the .NET Framework
  - .NET Framework is available directly
- Language innovations
  - Classes
  - Inheritance
  - Constructors
  - Structured exceptions
  - Optional stricter type checking

Visual Basic .NET
Feature Enhancements

- Structured exception handling
- Function overloading
- Free threading
- Namespaces
- Stricter type checking
- Parameterized constructors
- Class/Module statements

Visual Basic .NET
Additional .NET Changes

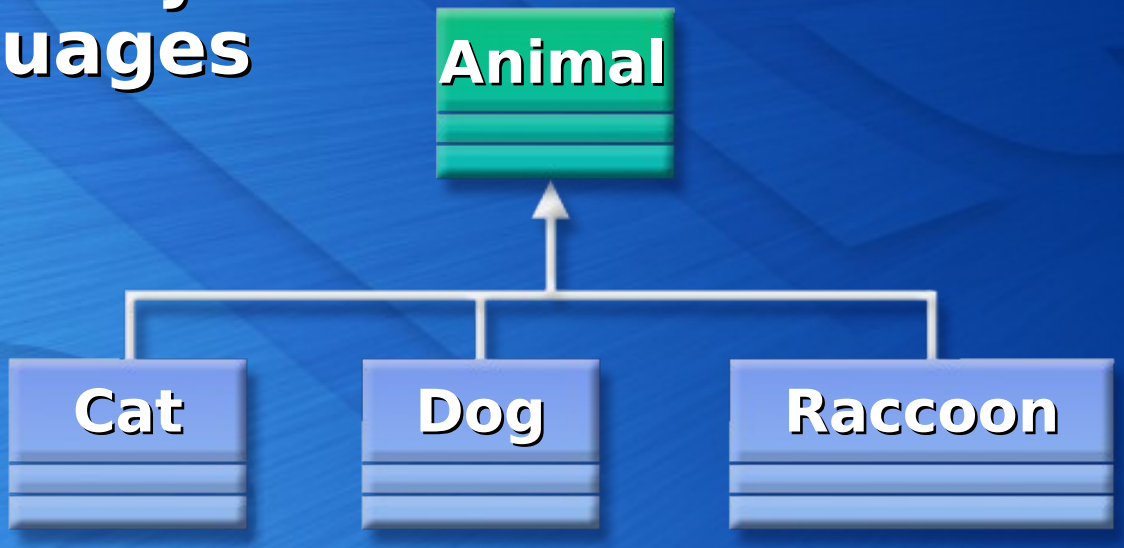
- Garbage collection
- Arrays are object-based
- Variant datatype replaced by object

Visual Basic .NET
Demo – Simple Visual Basic Application

- Show the basic steps to create Visual Basic .NET application

Visual Basic .NET
Inheritance

- Visual Basic is a full OOP player
- Overloading - Polymorphism
- Same object features as other Object-Oriented Languages



Visual Basic .NET
Structured Error Handling

- Try construct
  - Guards code between Try and first Catch statement
  - Nested Trys to resume behavior
- Catch statement(s)
  - Filters errors
- Finally block
  - Always executes
- Visual Basic .NET still supports “On Error GoTo”

Visual Basic .NET

On Error Versus Try/Catch

```
fReRaise = False
   On Error GoTo ErrHandler
<Code that may fail>
   GoTo Cleanup
ErrHandler:
if ConditionWeCanHandle Then
    <error handling code>
else
    fReraise = True
End if
CleanUp:
<clean up>
If fReraise Then _
    err.Raise err
```

```
Try
  <Code that may fail>
Catch
  ConditionWeCanHandle
   <error handling code>
Finally
  <clean up>
End Try
```

Visual Basic .NET
Demo

- Error Handling

Visual Basic .NET
Simplifications

- Eliminates default properties without parameters
- Only one form of assignment
- Always use parentheses around arguments

```
DoSomething(a,b)
```

- Fix variable declaration syntax:

```
Dim x, y as Integer
```

- New operator same for all classes

Visual Basic .NET
Reduce Programming Errors

- Parameterized constructors
- Variables can be initialized where declared
- Circular references are no longer a problem
  - Garbage collector recovers instances in reference cycles

Visual Basic .NET
Type Checking

- Option for stricter type checking
  - More errors detected during compilation
  - Prevents implicit conversions that may fail at run time
- Explicit conversion (casting) operator: CType
- Enables support for overloaded methods
- Array declarations must specify rank

Visual C#
Features

- First component-oriented language in the C/C++ family
- No header files, IDL, etc...
- Visual C# employs "definition at declaration" model
  - Much like Visual Basic, Pascal, Modula and Java
- Similar to Visual C++ "in-line" implementation
  - Without implication on code generation
- All code and declaration in one place
  - Keeps code consistent and maintainable
  - Much better to understand for team collaboration
  - Declaration accessible through metadata
- Conditional compilation but no macros

Visual C#
Statements

Flow Control and Loops

- if (<bool expr>) { ... } else { ... }
- switch(<var>) { case <const>: ...; }
- while (<bool expr>) { ... };
- for (<init>;<bool test>;<modify>) { ... }
- do { ... } while (<bool expr>);
- Structured error handling (Try/Catch)

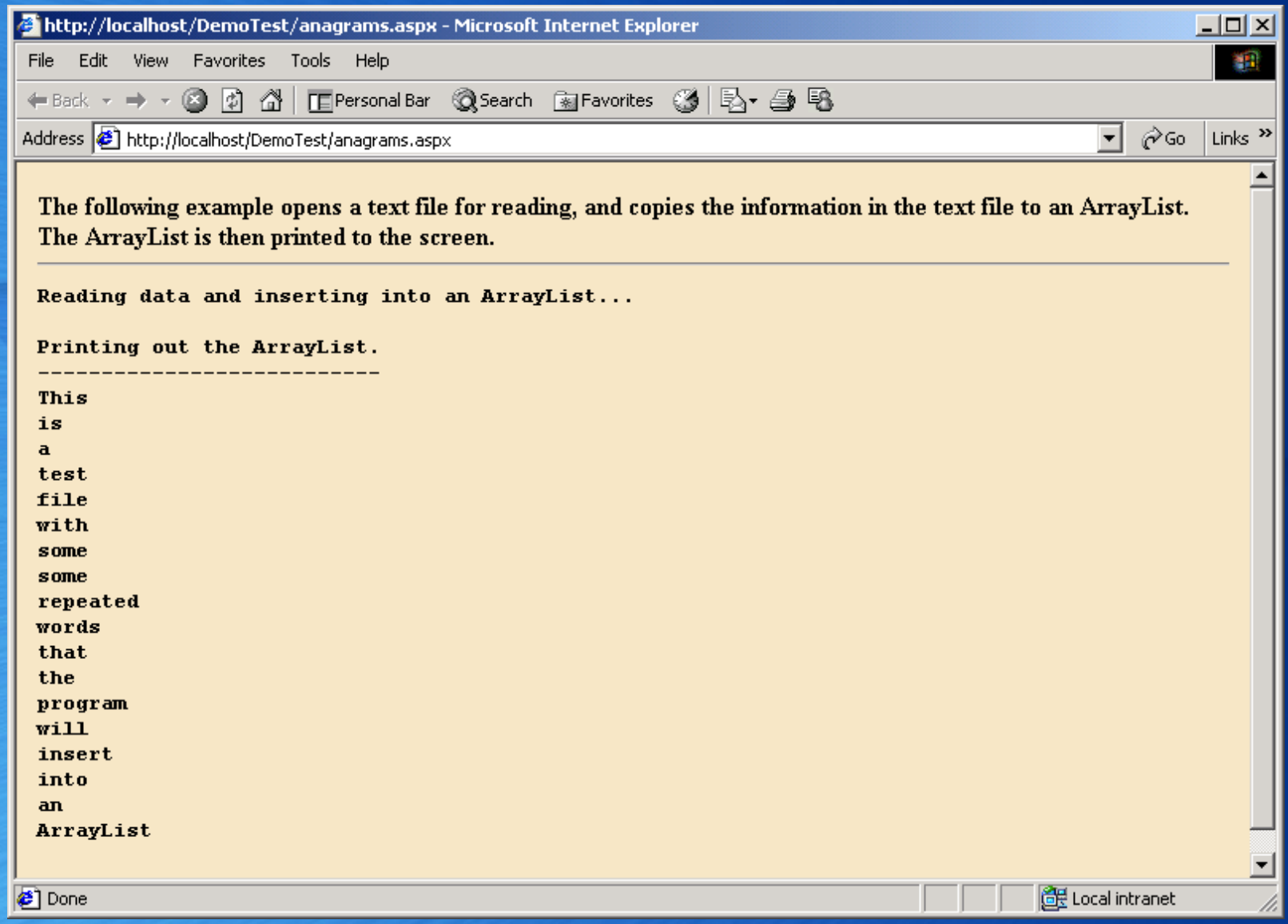
Visual C#
Collections

- Straight-forward support for iterating over collections
  - Can be used for arrays and other collections
- Can also be used with any custom class
  - Implements IEnumerable with GetEnumerator()
  - Returning object implementing IEnumerator

Visual C#
Operators

- Arithmetic: * / + - %
- Logical: & | ^
- Conditional: && ||
- Shift operators: << >>
- Relational: == != < > >= <= is
- Assignment: = *= /+ %= += -= <<= >>= &= |= ^=

Visual C#
Demo – Simple C# Applications



Visual C++ .NET

- Part of Visual Studio .NET IDE
- Visual C++ optionally targets .NET
  - Focus is on power, control and flexibility
  - No black boxes
  - Productive, but never at the expense of power and expressiveness
- ATL Server for native XML Web services
- Generates managed and unmanaged code

Managed Extensions To Visual C++

- It's still Visual C++
  - Minimal extensions in the ANSI spirit
  - Nothing taken away
  - Does everything Visual C# does
- Moving your existing code to the .NET Framework works
- Power and control
  - Mix managed and native code
  - Mix managed and unmanaged data
  - Allows you to gradually migrate
- Full access to the .NET Framework

Visual J#

- Java language tool for the .NET Framework
- Integrated into Visual Studio .NET
- Language Extensions
- Converters

Summary

- Select the language that is best for you
  - Visual Basic or VBScript developer → Visual Basic .NET
  - Visual C++ → Visual C# (Managed), Visual C#
  - JScript developer → JScript or Visual C#
  - Java developer→ Visual J#, Visual C#

Resources

- ***Basic Web Controls*** **- Basic syntax used to create Web controls**
  http://msdn.microsoft.com/library/dotnet/cpguide/cpconaspsyntaxforwebcontrols.htm
- ***Language Innovations in .NET*** **- A page with a little history, a little object-oriented programming, a little inheritance and even a little overloading**
  http://msdn.microsoft.com/vstudio/nextgen/technology/language.asp

Language Resources

- APL - http://www.dyadic.com/
- Cobol - http://www.adtools.com/info/whitepaper/net.html
- Component Pascal - http://www2.fit.qut.edu.au/CompSci/PLAS//ComponentPascal/
- Eiffel - http://dotnet.eiffel.com/
- Haskell - http://haskell.cs.yale.edu/ghc/
- Mercury - http://www.cs.mu.oz.au/research/mercury/
- Oberon - http://www.oberon.ethz.ch/lightning/
- Mondrian - http://www.mondrian-script.org
- Python - http://aspn.activestate.com/ASPN/NET/index
- Scheme - http://rover.cs.nwu.edu/~scheme/
- SmallTalk - http://www.qks.com/
- Standard ML - http://research.microsoft.com/Projects/SML.NET/index.htm

In-Depth Training On .NET Languages

Direct from Microsoft

- 2373A: Programming with Microsoft Visual Basic .NET (Prerelease)
- 2373B: Programming with Microsoft Visual Basic .NET
  - Available in Mar 2002
- 2124B: Introduction to C# for the Microsoft .NET Platform (Prerelease)
- 2559A: Introduction to Microsoft Visual Basic .NET Programming
  - Available in Mar 2002

In-Depth Training On .NET Languages

Direct from Microsoft

- 2558A: Programming with Microsoft Visual C++ .NET Managed Extensions
  - Available in May 2002
- 2609A: Introduction to Microsoft C# .NET Programming
  - Available in Jun 2002

Microsoft®

03.32